# Adaptive Bandwidth Control for Efficient Aggregate QoS Provisioning

Peerapon Siripongwutikorn[†] Sujata Banerjee[†‡] David Tipper[†]

***Abstract*—**
**This paper proposes an adaptive bandwidth control algorithm that efficiently provides aggregate loss guarantee to resolve the problem of inefficient bandwidth allocation due to incomplete, inaccurate traffic descriptors supplied by users. Because the control attempts to allocate the bandwidth only just enough to meet the QoS requirement, the amount of bandwidth saving compared to static allocation can be substantial. Another distinct advantage of our control algorithm is that no a priori information on the traffic characteristics of the aggregate is required. From the simulation study, the proposed control can maintain the packet loss QoS while attaining very high utilization, and is robust against different system configurations.**

**Symposium: High-Speed Networks**

## I. Introduction

Recently, aggregate traffic management and control is used to address the scalability issue in per-flow traffic treatment. This in turn raises another issue of how to guarantee statistical QoS in an aggregate level for high utilization. Static bandwidth allocation (see e.g., [1], [2], [3]) is inefficient because it assumes some specific stochastic model on the underlying traffic as well as accurate traffic parameter values. Such detailed information is unavailable because most of the time the input aggregate traffic can only be specified roughly in terms of the average rate or peak rate due to its unpredictable nature.

The motivation for this research is the need to guarantee the aggregate QoS under unknown input traffic characterization while being able to attain high link utilization. This research proposes to use Adaptive Bandwidth Control (ABC) on a per-hop basis to deliver the aggregate loss guarantee. The control essentially involves the bandwidth adaptation over time to ensure that the allocated bandwidth is just enough to maintain the specified loss requirement. As such, less bandwidth will be wasted due to overallocation.

Most existing work in ABC for loss guarantee so far has many shortcomings and there is room for improvement. Those based on the feedback control directly measure the packet loss or delay to adjust the service rate [4], [5]. The major issue in using feedback control is that they may fail to adapt in case of non-stationary or highly dynamic traffic conditions because the time period needed to acquire accurate loss statistic is too long. Another approach for ABC is based on traffic prediction which allocates bandwidth to match the predicted arrival rate, and hence results in negligible loss (see e.g., [6], [7], [8], [9]). For traffic with widely varying bit rate over time such as MPEG-coded video or VBR video, applying ABC has been shown to satisfactorily improve the network utilization. However, the resulting performance is difficult to quantify and control, and one may end up overallocating the bandwidth.

We develop an ABC algorithm based on fuzzy control to maintain the aggregate loss QoS. Our proposed ABC algorithm belongs to the class of feedback control algorithms. As mentioned before, the measured packet loss can be very inaccurate and thus is a poor feedback parameter. On the contrary, a statistic such as the average queue length has a much smaller variance and can be measured more accurately in a short time scale. Moreover, the increase in the average queue length necessarily implies higher loss, and vice versa. We therefore convert the target loss requirement to the target average queue length and use it as the control objective. We will demonstrate that adjusting the service rate to maintain the average queue length between two queue thresholds can keep the packet loss relatively constant. A unique aspect of our approach is that unlike any of the previous work, the short term loss is steadily maintained at the target level even in the face of dynamic non-stationary input traffic. Further, the control algorithm is computationally simple and requires only measuring only the long term average arrival rate, and the average queue length statistic. Such simplicity allows efficient implementation at very high-speed.

The remainder of this paper is organized as follows. In the next section, we present a fuzzy controller to maintain the packet loss steadily in a single queue, followed by a description of the performance metrics in §III. The performance of the proposed controller is evaluated in §IV on non-stationary long range dependence traffic with different control parameters to demonstrate its robustness. The conclusion will be provided in §V.

## II. Fuzzy Controller

In this section, we describe a fuzzy controller that maintains the average queue length in a single finite buffer queue. For a control system that cannot be adequately described by detailed mathematical equations, fuzzy control is a convenient way to synthesize a non-linear controller whose control law are heuristically derived through insight of the process under control. As stated earlier, we attempt to maintain $Q_k$ between some queue thresholds to achieve the loss requirement. This raises the problem of how to derive the queue thresholds from the loss requirement. As of the current stage of this work, we manually tune the thresholds to get the desired target loss. A more sophisticated mapping scheme will be left for further research.

In essence, a fuzzy controller consists of fuzzification, inference, and defuzzification steps. In the fuzzification step, a measured feedback value (called crisp input) is converted into

*This research is supported in part by the NSF CAREER award No. ANI-9702389.

[†]Telecommunication Program, University of Pittsburgh
[‡]Hewlett-Packard Laboratories

linguistic values (such as *low* or *high*), each of which represented by a fuzzy set. Each fuzzy set is associated with a membership function used to characterize how certain the crisp input belongs to the set. For a given crisp input, the membership function returns a real number in $[0, 1]$. The closer the membership function value is to 1, the more certain the input belongs to the set (e.g., more *low* or less *high*). A single crisp input value can take on more than one linguistic value if the domains of membership functions overlap as will be seen later. In the inference step, a set of rules called *rule-base*, which emulates the decision-making process of a human expert, is applied to the linguistic values of the inputs to infer the output (fuzzy) sets. These outputs are then defuzzified to the crisp output which represents the actual control signal for the process. We refer the reader to [10] for more complete background information on the fuzzy control.

The rest of this section describes the fuzzification, inference, and defuzzication steps of our controller. Let us first define the notation. A measurement period for the next control action (i.e., bandwidth adjustment) is denoted $T_m$, which corresponds to the control time scale. The $k$th time instant, denoted by $t_k$, refers to real time $t_k = kT_m$ and the $k$th interval refers to a time period starting from $t_{k-1}$ to $t_k$. Let $Q_k$ be the average queue length in packets (including the one in service) as *seen by the arrivals* measured over the $k$th interval.

Two state feedback parameters are used as the inputs to the fuzzy controller, which then computes the bandwidth to be allocated to serve the queue. The first feedback is the Exponential Weighted Moving Average (EWMA) of $Q_k$, denoted by $\hat{Q}_k$, which is given by

$$\hat{Q}_k = \alpha Q_k + (1 - \alpha)\hat{Q}_{k-1} \tag{1}$$

The EWMA operation acts as a low-pass filter on $Q_k$ to smooth out noise in measurement. The other feedback parameter is the normalized change in the EWMA queue length ($\Delta\hat{Q}_k$), given by

$$\Delta\hat{Q}_k = \frac{\hat{Q}_k - \hat{Q}_{k-1}}{u_{th} - l_{th}} \tag{2}$$

Let $C_{k+1}$ denote the allocated bandwidth during the time interval $(t_k, t_{k+1}]$. Based on $\hat{Q}_k$ and $\Delta\hat{Q}_k$, $C_{k+1}$ is determined at $t_k, k = 1, 2, \ldots$, and remains constant during $(t_k, t_{k+1}]$.

### A. Fuzzification

Fuzzification is the process of translating crisp inputs for each input variable $i$ into linguistic values. We define $N_i$ linguistic values $A_i^{(m)}, m = 1, 2, \ldots, N_i$ as well as their membership functions. For $\hat{Q}_k$, its linguistic values $A_1^{(m)}, m = 1, 2, 3,$ are (i) low, (ii) medium, and (iii) high, with the corresponding membership functions $G_Q^{(m)}(\hat{Q}_k)$ shown in Fig. 1. Besides a triangular shape, many other choices for the shape of the membership functions also exist, including trapezoidal, Gaussian, and etc. However, the triangular shape is a standard choice used in most industry applications due to its simple expression. The membership functions are defined such that $\sum_{m=1}^{3} G_Q^{(m)}(\hat{Q}_k) = 1, \forall \hat{Q}_k \in [0, K]$ where $K$ is the buffer size in packets. Note that a single value of $\hat{Q}_k$ can take on

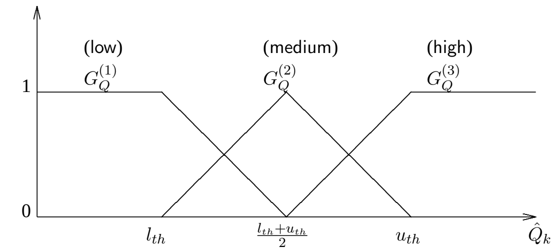



Fig. 1. Membership functions for the $\hat{Q}_k$

more than one linguistic value. For instance, if $(l_{th} + u_{th})/2 < \hat{Q}_k < u_{th}$, $\hat{Q}_k$ can be both medium and high but with different degrees of certainty indicated by the outputs of the membership functions.

Similarly for $\Delta\hat{Q}_k$, its linguistic values $A_2^{(m)}, m = 1, 2, 3, 4, 5$, are (i) -high, (ii) -low, (iii) zero, (iv) +low, and (v) +high, with their membership functions $G_\Delta^{(m)}(\Delta\hat{Q}_k)$ shown in Fig. 2. As before, the membership functions are defined such that $\sum_{m=1}^{5} G_\Delta^{(m)}(\Delta\hat{Q}_k) = 1, \forall \Delta\hat{Q}_k \in \mathbb{R}$. Here, the member-

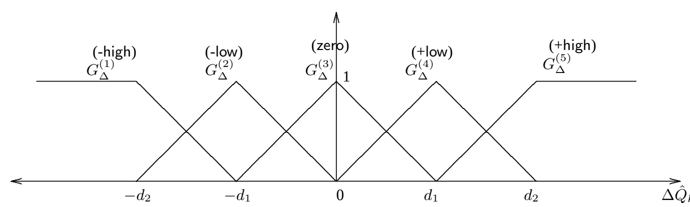



Fig. 2. Membership functions for the $\Delta\hat{Q}_k$

ship functions of $\Delta\hat{Q}_k$ have two tunable parameters: $d_1$ and $d_2$. In this paper, these two parameters are respectively set to 0.4 and 0.8. We show later that fairly good control results are obtained without any further tuning of the membership functions.

### B. Inference and Defuzzification

After the crisp inputs are mapped to the linguistic values through the membership functions in the fuzzification step, inference rules are applied to determine the output by using a rule-base. The rule-base is a set of rules that emulates the decision-making process of the human expert controlling the system. The rule is written in the form

IF *premise* THEN *consequent/action*

where *premise* is a combination of input linguistic values and *consequent* is an action to be taken. Because there are three linguistic values for $\hat{Q}_k$ and five for $\Delta\hat{Q}_k$, the total number of rules is 15. If the premise is true, we call the rule as being *active*. In our case, the rule-base is in a form called functional fuzzy system where each rule $i$ is written down as:

Rule $i$: IF $\hat{Q}_k$ is $A_1^{(l)}$ and $\Delta\hat{Q}_k$ is $A_2^{(m)}$,
THEN $b_i = G_Q^{(l)}(\hat{Q}_k) * G_\Delta^{(m)}(\Delta\hat{Q}_k) * x_i$

where $x_i$ is the service rate adjustment associated with rule $i$. As a result, $b_i$ is the change in the service rate if rule $i$ is active, with the term $G_{Ql}(\cdot) * G_{\Delta m}(\cdot)$ being the weight. The values of the $x_i$'s are established based on insights of the queue behavior. For example, if $\hat{Q}_k$ is low and $\Delta\hat{Q}_k$ is zero, then $x_i$ should be some small negative number in order to decrease the service rate and hence would likely increase $\hat{Q}_{k+1}$. Intuitively, $x_i$ should be related to the long-term average rate of the input traffic, which can be easily obtained from on-line measurement. Their values selected for use in the simulation study are tabulated in Table I as the percentage of the long-term input rate. In general, $x_i$'s can be made adaptive but they are fixed at this stage of the work. Once $b_i$'s have been determined from the inference step, the defuzzification is performed to obtain $C_{k+1}$ by using $C_{k+1} = C_k + \sum_{i=1}^{15} b_i$.

| $\hat{Q}_k$ | $\Delta\hat{Q}_k$ | | | | |
|---|---|---|---|---|---|
| | -high | -low | zero | +low | +high |
| low | -10% | -5% | -2.5% | 0 | 0 |
| medium | -4% | -1% | 0 | 1% | 4% |
| high | 1% | 1% | 2% | 5% | 10% |

TABLE I

VALUES OF $x_i$ FOR RULE-BASE (AS A FRACTION OF THE LONG TERM AVERAGE RATE)

We have not attempted to tune our fuzzy controller to provide optimal performance because this appears to be very difficult due to the many degrees of freedom associated with the membership functions, rule-base, and the parameters thereof. However, as we show later, any further tuning beyond the basic intuitive ideas is not necessary and the fuzzy controller performs well. Tuning the fuzzy controller to provide optimal performance will be the subject of future research.

## III. PERFORMANCE METRICS

Primary performance metrics considered are the bandwidth utilization and the average short term packet loss. The utilization is simply a fraction of the server busy period determined over a given operating period of the system. For the loss performance index, we consider the time-average of the short term packet loss. Here we define the short-term loss as the one evaluated from the minimum number of observations ($N$) required to obtain $r$% relative precision on the $100(1-\alpha)$% confidence interval of the target loss probability ($p_{\text{loss}}$). In particular, $N \geq z_{1-\alpha/2}\frac{p_{\text{loss}}(1-p_{\text{loss}})}{(\frac{r}{100}p_{\text{loss}})^2}$, where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of a unit normal variate [11, p.217]. For $p_{\text{loss}} = 10^{-3}$, $r = 5\%$, and $95\%$ confidence interval, $N$ is $2.17 \times 10^6$ packets. The relative precision of $5\%$ at $95\%$ confidence interval is used in our simulation study presented next. Then the approximated measurement period ($W$) for the short term loss is $\frac{N}{\lambda}$, where $\lambda$ is the long-term average packet arrival rate. Let $\epsilon_k$ denote the short term loss observed over the past $W$ seconds at the loss measurement time instant $t_k$ i.e., $(\max(0, t_k - W), t_k]$. Assuming that the measurement instants also move by $T_m$ seconds. That is, we measure the loss over moving windows of length $W$ that are shifted by $T_m$ at the time. Then, for the system operating period of length $T$, we define the loss performance index as

$$\hat{\varepsilon} = \frac{1}{T}\sum_{k=1}^{\lfloor T/T_m \rfloor} \epsilon_k, \tag{3}$$

which captures the short term loss behavior. The coefficient of variation (C.o.V), $\mathrm{Var}(\epsilon_k)/\hat{\varepsilon}$, will also be used to indicate the fluctuation of the short term loss.

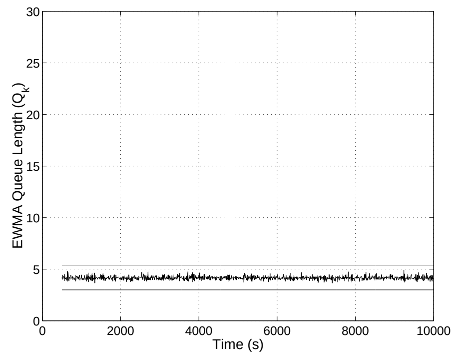
## IV. EXPERIMENTAL RESULTS

In this section, we study performance of the fuzzy controller on Long Range Dependence (LRD) aggregate traffic constructing from a number of identical Pareto on-off flows. To model a highly dynamic condition, we make the aggregate non-stationary in the sense that flows in the aggregate arrive and depart over time. The Pareto on-off source alternates between on and off states. During the on state, packets are generated at a constant rate $R$ packets/second and the number of generated packets has a Pareto distribution with mean $P$. During the off state, the source stays idle for a Pareto distributed length of time. The mean off periods used is 2 seconds. The shape parameter ($\gamma$) of 1.5 for the Pareto distribution is used. An aggregate of such Pareto on-off sources contributes LRD traffic with the Hurst parameter $H = (3-\gamma)/2$ [12]. The values $P = 192$ and $R = 160$ are used, corresponding to the source average rate of 60 packets/sec. The packet size is exponentially distributed with mean 53 bytes. The source arrival process in the connection level is Poisson with rate 2 per second and has an exponential holding time of 100 seconds. This is equivalent to 200 active sources inside the aggregate on the average, and the long-term aggregate average rate of 12,000 packets/sec.

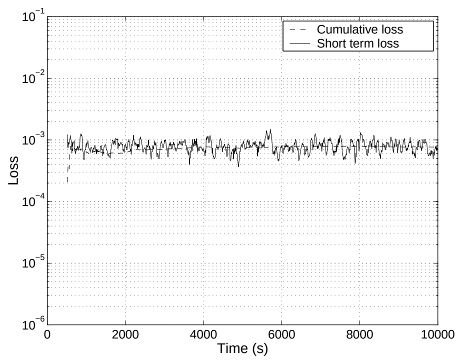
We consider $p_{\text{loss}} = 10^{-3}$, the buffer size ($K$) of 30 and 1024. The controller has three adjustable parameters – the EWMA weight $\alpha$, the control time scale $T_m$, and the threshold pair $l_{th}$,$u_{th}$. The performance will be evaluated under $\alpha$ of 0.1 and 0.3, and $T_m$ of 0.5, 1, and 5 seconds. From a rough offline tuning, we found that a threshold pair $l_{th}$ = 10% and $u_{th}$ = 18% of the buffer size approximately yields the desired $p_{\text{loss}}$ at $K = 30$. To provide comparative evaluation, we compare our results to optimal static allocation and Equivalent Bandwidth (EB) allocation.

- In the optimal static allocation, we determine by trial-and-error through simulation the (approximated) optimal, i.e., smallest amount of static bandwidth required to attain the *cumulative loss* at the given $p_{\text{loss}}$. Such optimal value is infeasible to obtain in practice and is given here as a best case scenario.
- For the EB allocation, we use the EB formula for LRD traffic presented in [13], in which case the bandwidth is reallocated based on the EB formula at every flow arrival and departure instants. Note also that the use of EB allocation is a hypothetical scenario where the flow arrival and departure process within the aggregate is assumed known.

In either cases, if the fuzzy control can achieve more or less the same amount of allocated bandwidth, it is considered superior in that no knowledge of traffic characteristics is required.

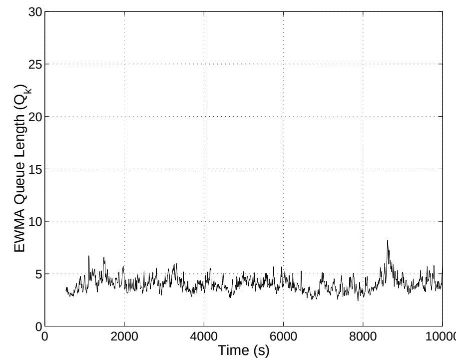



(a) Average Queue Length

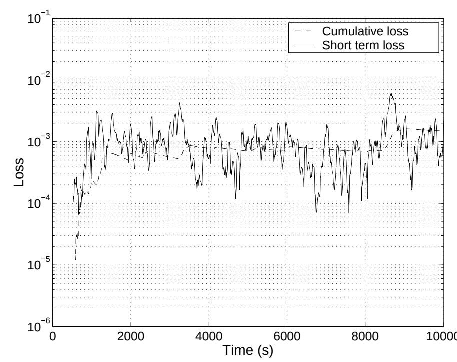



(b) Loss

Fig. 3. Fuzzy control ($K = 30$, $T_m = 1$ second,$\alpha = 0.1$)




(a) Average Queue Length




(b) Loss

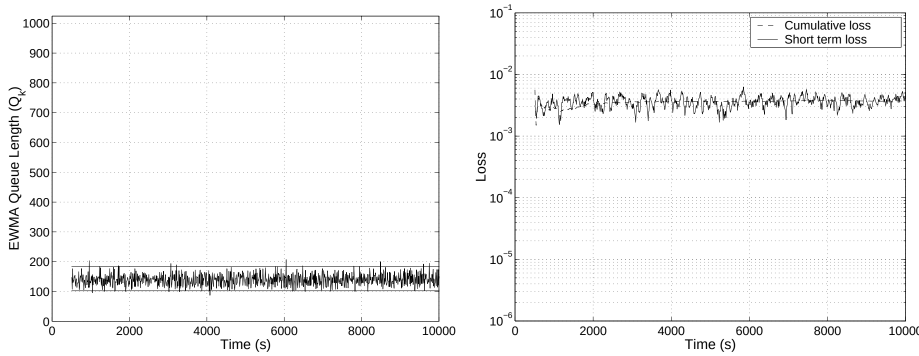
Fig. 4. Optimal static allocation ($K = 30$)

Fig. 3 and 4 show the sample paths of the short-term loss and $Q_k$ averaged across five runs for the proposed fuzzy control and the optimal static allocation at $T_m = 1$ second. We found that the average allocated bandwidth from the fuzzy control are less than 1% different from the optimal value for every $T_m$ and $\alpha$ used. With the static allocation, however, the short term loss cannot be maintained steadily as shown in the figure. This can be undesirable because the variation in short term loss can adversely affect QoS-sensitive transport of real-time traffic [14]. In addition, this kind of optimal bandwidth allocation is impossible to determine in real-time. Note also that the behavior of the average queue length and packet loss are directly related. By increasing $K$ to 1024, the fuzzy control still performs relatively well in maintaining the average length and hence the short term loss (Fig. 5) even though the resulting loss is somewhat higher than the desired value. This deviation can be addressed by tuning the thresholds to the right values, which is our ongoing research work.

Table II provides the performance comparisons among the fuzzy control, optimal static allocation, and EB allocation at different buffer sizes and control parameters. All the number shown are averaged from five runs and the short-term packet loss as well as the utilization have the relative precision of less than 5% at 95% confidence interval*. In every cases, the EB method overallocates the bandwidth and results in zero packet loss. Compared to the EB allocation, the fuzzy control introduces the bandwidth saving around $35\%$, as seen from much higher utilization.

We can infer some properties of our algorithm from the simulation results shown in Table II. First, at a given buffer size, the average of short term loss is robust over a wide range of $T_m$ but its variation is not. In particular, higher variation (C.o.V) in the short term loss is observed as $T_m$ increases, which is intuitive because the bandwidth is adjusted too slow. Using smaller $T_m$ enables finer control and hence better performance. However, the choice of $T_m$ is not easily chosen because too small $T_m$ will lead to bandwidth thrashing whereby the effect of bandwidth changes has not reflected in the average queue length performance. This implies that there should be some optimal value of $T_m$ to use. Nevertheless, the results indicate that our control algorithm still performs relatively well across a wide range of the control time scale. Second, for a given target packet loss, the thresholds does not increase linearly with the buffer size. In this case, $l_{th} = 10\%$ and $u_{th} = 18\%$ of the buffer size ($K$) yields the desired loss at $K$ = 30 but not $K$ = 1024 packets. This calls for a more sophisicated mapping between the loss requirement and the queue thresholds. Finally, the control performance appears insensitive to the EWMA weight $\alpha$. By increasing $\alpha$, the EWMA queue length will change more rapidly, which should provide the controller more faster response to traffic dynamics. However, the results indicate no obvious difference for different values of $\alpha$, which facilitates the parameter selection in practice.

*Except in case of static optimal allocation at $K$ = 1024, which requires 20 runs.



(a) Average Queue Length (b) Loss

Fig. 5. Fuzzy control ($K = 1024$, $T_m = 1$ second, $\alpha = 0.1$)

| Parameters | | | Fuzzy Control | | | Optimal Static | | | EB |
|---|---|---|---|---|---|---|---|---|---|
| | | | Avg.Loss ($\hat{\varepsilon}$) | $\mathrm{Var}(\epsilon_k)/\hat{\varepsilon}$ | Util | Avg.Loss ($\hat{\varepsilon}$) | $\mathrm{Var}(\epsilon_k)/\hat{\varepsilon}$ | Util. | Util. |
| $K$ = 30 | $T_m = 0.5$ | $\alpha = 0.1$ | 0.84$e$-3 | 0.43 | 0.812 | 1.04$e$-3 | 2.00 | 0.796 | 0.380 |
| | | $\alpha = 0.3$ | 0.70$e$-3 | 0.43 | 0.814 | | | | |
| | $T_m = 1$ | $\alpha = 0.1$ | 0.79$e$-3 | 0.47 | 0.812 | | | | |
| | | $\alpha = 0.3$ | 0.68$e$-3 | 0.48 | 0.813 | | | | |
| | $T_m = 5$ | $\alpha = 0.1$ | 0.86$e$-3 | 0.82 | 0.808 | | | | |
| | | $\alpha = 0.3$ | 0.68$e$-3 | 0.56 | 0.812 | | | | |
| $K$ = 1024 | $T_m = 0.5$ | $\alpha = 0.1$ | 3.65$e$-3 | 0.40 | 0.860 | 1.10$e$-3 | 3.61 | 0.862 | 0.488 |
| | | $\alpha = 0.3$ | 2.49$e$-3 | 0.44 | 0.895 | | | | |
| | $T_m = 1$ | $\alpha = 0.1$ | 3.66$e$-3 | 0.45 | 0.877 | | | | |
| | | $\alpha = 0.3$ | 3.75$e$-3 | 0.44 | 0.903 | | | | |
| | $T_m = 5$ | $\alpha = 0.1$ | 3.66$e$-3 | 0.73 | 0.899 | | | | |
| | | $\alpha = 0.3$ | 5.33$e$-3 | 0.60 | 0.904 | | | | |

TABLE II

PERFORMANCE COMPARISONS OF FUZZY CONTROL WITH OPTIMAL STATIC ALLOCATION AND EQUIVALENT BANDWIDTH

## V. CONCLUSION

We address the problem of bandwidth allocation to guarantee the aggregate loss QoS by using adaptive bandwidth control. The control adapts the allocated bandwidth such that the loss QoS is maintained while being able to achieve high utilization. A major appealing aspect of the control is that it does not require the knowledge of traffic characteristics. We develop a simple fuzzy logic controller and evaluate its performance under a highly dynamic traffic condition. The proposed control allocates the bandwidth that is slightly lower than the optimal static allocation and introduces significant amount of bandwidth saving compared to equivalent bandwidth allocation. Further, the controller is shown to be robust against different control parameters, rendering it tolerant to suboptimal parameter selections. We are aware that our preliminary results with only one traffic type may not be used to generalize the controller performance, and hence we plan to experiment on other different kinds of traffic mixes in the aggregate and real traffic traces. In addition, two major issues of control time scale and the queue thresholds selection require further investigation.

## REFERENCES


[1] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 968–981, Sept. 1991.

[2] N.G. Duffield and N. O´Connell, "Large Deviations and Overflow Probabilities for the General Single-Server Queue, with Applications," in *Math. Proc. Cambridge Phil. Soc.*, 1996.

[3] C. Courcoubetis, V.A. Siris, and G.D. Stamoulis, "Application and Evaluation of Large Deviation Techniques for Traffic Engineering in Broadband Networks," in *ACM SIGMETRICS'98, Madison, WI, USA*, 1998.

[4] I. Hsu and J. Walrand, "Dynamic Bandwidth Allocation for ATM Switches," *J. of Applied Probability*, vol. 33, no. 3, pp. 758–771, 1996.

[5] S. Rampal, D. Reeves, Y. Viniotis, and D. Argrawal, "Dynamic Resource Allocation Based on Measured QoS," Tech. Rep. Technical Report TR 96-2, Center for Advanced Computing and Communication, North Carolina State University, Jan. 1996.

[6] S. Chong, S.-Q. Li, and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[7] A.M. Adas, "Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 635–645, Oct. 1998.

[8] J.R. Gallardo, D. Makrakis, and M. Angulo, "Dynamic Resource Management Considering the Real Behavior of Aggregate Traffic," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 177–185, June. 2001.

[9] Z. Sahinoglu and S. Tekinay, "A Novel Approach Bandwidth Allocation: Wavelet-Decomposed Signal Energy Approach," in *Proc. IEEE GLOBECOM'01*, San Antonio, TX, Dec. 2001.

[10] K.M. Passino and S. Yurkovich, *Fuzzy Control*, Addison Wesley, 1998.

[11] R. Jain, *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, Inc., 1991.

[12] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 71–86, Feb. 1997.

[13] I. Norros, "On the use of Fractional Brownian Motion in the Theory of Connectionless Networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 953–962, Aug. 1995.

[14] K. Park and W. Willinger, *Self-similar Network Traffic and Performance Evaluation*, chapter 1, p. 10, Wiley-Interscience, 2000.